CSCE 421: Spring 2023 Homework 2

Due on Thur, Feb 16, by 11:59 PM.

Submit your assignments (Code + Report) separately on gradescope. Please name your coding assignment as assignment2.py. Use the provided python template i.e, complete ONLY the functions (DO NOT edit function definitions, code outside the function, or use any other libraries).

A Few Notes:

- You may submit a single Python script as well as a written report generated by LaTex to both compose the written portions and the code together.
- Coding assignments are primarily designed for Python.
- Please start early!!
- If you need to use a different editor to write the solutions - please contact the TA and Instructor first.
- This is an individual assignment. While you are welcome to discuss general concepts together and on the discussion board your solutions must be yours and yours alone.
- **SHOW YOUR WORK.**

Problem 1: Coding: Linear regression implementation.

In this question, we will learn how to implement a linear regression model.

LinearRegression/train.csv is your training data and LinearRegression/test.csv is your test data.

Your submission should include a script which can be run seamlessly and performs all the following steps one after another. Any submission with a runtime error would result in lost points.

(1) Download and read the data. You may use *pandas* library and use *read csv* function.

(2) Print the data. How does the data look like? Add a short description about the data to the report file. (You may use *head()* function in *pandas* library)

(3) Prepare your input data and label. (Note: find and remove Nan samples).

(4) Implement LinearRegression class. There are four functions in this class: __init__ (initialization, you don't need to work on this), fit(), update_weights(), and predict(). You need to decide what variables should be passed to each function, and finish the rest three functions.

(5) Build your model and train with training set.

(6) Make predictions with test set.

(7) Calculate and return the mean square error of your prediction.

(8) Plot your prediction and labels. Add the plot to the report file. (You may use *matplotlib* package.)

Problem 2: Coding: Preprocessing the Data.

In this question, we will learn about data preprocessing. More specifically, we want to prepare the data so that it will be ready for being fed into a machine learning model. Generally, the data contains missing values and also has categorical (non-numerical) values. We will learn how to prepare such data.

The dataset we will use in this assignment is called *Hitters* (can be downloaded from https://github.com /jcrouser/islr-python/blob/master/data/Hitters.csv).

Your submission should include a script which can be run seamlessly and performs all the following steps one after another. Any submission with a runtime error would result in lost points.

You may use libraries and you do not need to implement anything from scratch.

(1) Download and read the data. For Python, you may use *pandas* library and use *read csv* function

(2) Print the data. How does the data look like? Add a short description about the data to the report file. (You may use *head()* function in *pandas* library)

(3) Return the shape of the data. Shape means the dimensions of the data. (In Python, *pandas* dataframe instances have a variable *shape*)

(4) Does the data have any missing values? How many are missing? Return the number of missing values. (In *pandas*, check out *isnul()* and *isnul()*.sum())

(5) Drop all the rows with any missing data. (In *pandas*, check out *dropna()*. *dropna()* accepts an argument *inplace*, check out what it does and when it comes in handy.)

(6) Extract the features and the label from the data. Our label is *NewLeague* and all the others are considered features.

(7) Data preprocessing. We want to do one-hot encoding for categorical features. To do so, we first need to separate numerical columns from nonnumerical columns. (In *pandas*, check out $.select\_dtypes(exclude = ['int64','float64'])$ and $.select\_dtypes(include = ['int64','float64'])$. Afterwards, use *get dummies* for transforming to categorical. Then concat both parts (*pd.concat()*).

(8) Transform the output into numerical format. If you have selected the label as a *pandas* series, you can use *.replace()* function. In the label, transform *'A'* to 0 and *'N'* to 1.

Problem 3: Models for Hitters

In this question, we will apply simple classification algorithms, linear regression and logistic regression, to our preprocessed data, completing our first machine learning pipeline.

This problem comes after Problem 2 so the input data should be the one you have prepared for that.

Your submission should include a script which can be run seamlessly and performs all the following steps one after another. Any submission with a runtime error would result in lost points.

You may use libraries and you do not need to implement anything from scratch.

(1) **Prediction**: Using 80% of the data as a training set and 20% as a testing set, please train a linear regression model and a logistic regression model.

(2) (a) Please provide the coefficients for each feature for both models. Are they the same? Are they different? Why? Please describe your observation in the report file.

(2) (b) Please plot the ROC curve for both models. What are the area under the curve measurements? Add the ROC curves to the report file.

(2) (c) What is the optimal decision threshold to maximize the f1 score? Include the optimal threshold into the report file. How did you calculate the optimal threshold?

(3) **Five-fold Cross-validation**: Repeat (1) using a stratified, five-fold cross-validation.

(3) (a) Do the features change in each fold? Please explain in the report file.

(3) (b) Please provide a mean and 95% confidence interval for the AUROCs for each model.

(3) (c) Please provide a mean and 95% confidence interval for the f1 score for each model.